🛠️ Ejercicio Práctico

📌 **Título:** Detección de vulnerabilidad XSS mediante fuzzing con Burp Suite

---

🎯 Objetivo:

Simular una auditoría ética en un formulario web vulnerable utilizando **Burp Suite**. El propósito es **detectar una posible vulnerabilidad de tipo XSS** utilizando **fuzzing automatizado** con el módulo **Intruder**, e interpretar los resultados para determinar si el sistema es vulnerable.

🧱 Escenario:

El sitio de pruebas `http://testphp.vulnweb.com` contiene un formulario de búsqueda accesible en la ruta `/artsearch.php`.
El parámetro `artname` no cuenta con filtrado adecuado, y se sospecha que podría permitir **inyección de scripts (XSS)**.

🧪 Actividades:

✅ Parte 1 – Interceptar y preparar la solicitud

1. Abre **Burp Suite** y configura tu navegador con proxy en `127.0.0.1:8080`.
2. Accede a `http://testphp.vulnweb.com/artsearch.php`.
3. En el formulario de búsqueda, escribe cualquier palabra (ej. `test`) y haz clic en buscar.
4. Intercepta la solicitud generada usando **Proxy > Intercept**.
5. Envíala al módulo **Intruder** con clic derecho → “Send to Intruder”.

✅ Parte 2 – Configurar fuzzing en Intruder

1. En **Positions**, marca el valor del parámetro `artname` para ser reemplazado (`§test§`).

2. En la pestaña **Payloads**, carga una lista de payloads XSS comunes. Ejemplos:

```
<script>alert(1)</script>
"><img src=x onerror=alert(1)>
<svg/onload=alert('XSS')>
'"><svg onload=confirm(1)>
```

3. Haz clic en **Start attack**.

✅ Parte 3 – Analizar resultados

1. Revisa las respuestas de la aplicación (columnas “Status”, “Length”, “Response”).

2. Observa si alguna respuesta muestra el contenido inyectado reflejado.

3. Usa **Render** en la pestaña Response para visualizar si el código se ejecuta.

📋 Entregables esperados:

1. Captura de pantalla de la configuración en Intruder.

2. Captura de respuesta donde se confirma o sospecha ejecución de payload XSS.

3. Explicación de por qué el sistema es vulnerable o no.

4. Recomendaciones de mitigación para el equipo de desarrollo.

🧠 Preguntas de reflexión:

- ¿Por qué los ataques XSS reflejados son peligrosos para usuarios finales?
- ¿Qué importancia tiene codificar correctamente la salida en aplicaciones web?
- ¿Qué limitaciones tiene el análisis automático frente al análisis manual?
